TITLE OF THE INVENTION

INFORMATION PROCESSING APPARATUS, ITS CONTROL METHOD AND CONTROL PROGRAM

FIELD OF THE INVENTION

The present invention relates to an information processing apparatus, its control method and a control program.

BACKGROUND OF THE INVENTION

In an information processing apparatus connectable to an external device such as a digital still camera, e.g., a printer or printer-based multifunction product with a digital still camera connection function, the types of connectable external devices connectable to the information processing apparatus are limited.

In such case where connectable combinations between the external devices and the information processing apparatus are limited, a user as an operator must obtain an external device such as a digital still camera and an information processing apparatus such as a printer or a printer-based multifunction product with a digital still camera connection function, corresponding to each other, at the same time.

Further, in a case where an external device such as a digital still camera with a new function is

obtained, if the information processing apparatus such as a printer with a digital still camera connection function is old in comparison with the external device, the new function of the digital still camera cannot be fully performed, otherwise, in order to fully use the new function, it is necessary to prepare a new information processing apparatus such as a printer corresponding to the new function.

On the other hand, vendor-specific information on the peripheral device side can be transmitted to a PC without generating a vendor-specific PC driver by embedding the vendor-specific information (e.g., a media ID of a memory card) in general standard information and transmitting the vendor-specific information to the PC. However, the amount of transmittable information is limited, and further, vendor-specific operation control cannot be performed.

In this manner, conventionally, there is a limitation on the types of external device having a function which can be fully performed when the device is connected to an information processing apparatus, and the information processing apparatus and the external device cannot be effectively utilized.

SUMMARY OF THE INVENTION

Accordingly, the present invention enables

control of external device in a general-purpose manner, and enables control with an expanded function in correspondence with an external device. More particularly, the present invention provides an information processing apparatus connectable with an external device, comprising: a first driver storage unit to store a first driver to control a connected external device in a general-purpose manner; determination means for determining whether or not a second driver to control the connected external device in a device-specific manner exists in the external device; an acquisition unit to, if it is determined that the second driver exists, obtain the second driver from the connected external device; and second driver storage means for storing the obtained second driver.

Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same name or similar parts throughout the figures thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate embodiments of the invention and, together

with the description, serve to explain the principles of the invention.

Fig. 1 is a block diagram showing a schematic construction of an information processing apparatus 100 according to an embodiment of the present invention;

Fig. 2 is an example of memory mapping in a ROM 102 and an image memory 104 of the information processing apparatus 100, according to the embodiment of the present invention;

Fig. 3 is a flowchart showing an example of processing in a case where a USB device unit is connected to the information processing apparatus 100, according to the embodiment of the present invention; and

Fig. 4 is a flowchart showing an example of processing for device-specific driver in the information processing apparatus 100, according to the embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will now be described in detail in accordance with the accompanying drawings.

First, an information processing apparatus 100 will be described in detail. In the present embodiment, the information processing apparatus 100 includes a

printer or printer-based multifunction product with a connection function for external device such as a digital still camera. Further, an external device includes a digital still camera, a cellular phone or a PDA (Personal Digital Assistance) with an image sensing function. Fig. 1 is a block diagram showing a schematic construction of the information processing apparatus 100 according to an embodiment of the present invention.

In the information processing apparatus 100, a CPU 101 is a system controller which controls the overall information processing apparatus 100.

A ROM 102 holds fixed data such as control programs executed by the CPU 101, a data table and an operating system (OS) program. In the present embodiment, the various control programs stored in the ROM 102 perform software execution control such as scheduling, task switching and interruption processing under the control of the OS stored in the ROM 102, thus realize a multitask function for print control, reading control, communication control and the like. Further, a general-purpose driver to control an external device connected to a USB A connector 117, e.g., a general-purpose digital still camera, is also stored in the ROM 102. The CPU 101 operates in accordance with the general-purpose driver program, to control the external device connected to the USB A connector 117.

A RAM 103 comprises, e.g., an SRAM (Static Random Access Memory) requiring a backup power source. Data are held on the RAM 103 with a primary battery (not shown) for data backup. The RAM 103 holds program control variables and the like which must not be deleted. Further, the RAM 103 has a memory area for storing operator-registered values and management data for the information processing apparatus 100.

An image memory 104 comprises, e.g., a DRAM (Dynamic Random Access Memory). The image memory 104 mainly holds image data handled by the information processing apparatus 100, print data to be transmitted to a printing unit 113 (to be described in detail later), status information obtained from the printing unit 113 and the like. Further, a part of the image memory 104 can be ensured as a work area for execution of software processing. Further, a device-specific driver to control an external device connected to the USB A connector 117, i.e., a digital still camera, in an expanded manner, is also stored in the image memory 104. The CPU 101 operates in accordance with the device-specific driver program, to control the external device connected to the USB A connector 117 in an expanded manner.

A data converter 105 performs interpretation of PDL (Page Description Language) or the like, and performs image data conversion such as CG (Computer

Graphics) mapping of character data.

In a reading controller 106, a reading unit 107 optically reads an original document with a CIS image sensor (contact image sensor), then an image signal, converted as electric image data, is subjected to various image processings such as binarization and halftone processing via an image processing controller (not shown), and high-precision image data is outputted. Note that in the present embodiment, the reading controller 106 and the reading unit 107 are available for sheet reading control for reading with a fixed CIS image sensor while conveying an original, and also available for book reading control for scanning an original fixed on a platen with a moving CIS image sensor. The image sensor may be a CCD image sensor.

An operation display unit 108 has an operation unit with various keys, e.g., numeral value input keys, character input keys, single-touch telephone number keys, a mode setting key, a determination key, cancellation key and the like, for determination of image-transmission destination data and registration of preset data, an LED (light emitting diode) and an LCD (liquid crystal display) and the like. The operator's various input operations, display of operation status of the information processing apparatus 100 and the like are made at the operation display unit 108. Further, when a warning is made to the operator, the

LED flashes and warning information is displayed on the LCD.

A communication controller 109 comprises a MODEM (modulator-demodulator), an NCU (Network Control Unit) and the like. In the present embodiment, the communication controller 109 is connected to an analog communication line (PSTN) 131 for line control in T-30 protocol communication, call origination and call in to the communication line.

A resolution conversion processor 110 performs resolution conversion control such as mutual image data conversion between milli-based image data and inch-based image data. Note that in the resolution conversion processor 110, image data scaling processing is also available.

A coding/decoding processor 111 performs mutual coding/decoding or scaling on image data (non-compressed, MH, MR, MMR, JBIG, JPEG etc.) handled by the information processing apparatus 100. For example, upon printing based on image data from an external device connected to the USB A connector 117, e.g., a digital still camera, the coding/decoding processor 111 performs decoding processing on a JPEG file received from the digital still camera.

A printing controller 112 performs various image processings, such as smoothing processing, printing-density correction processing and color correction, on

image data to be print-outputted, via an image processing controller (not shown), thereby converts the image data to high-precision image data and outputs the data to the printing unit 113 (to be described in detail later).

The printing unit 113 is a printing apparatus such as a laser beam printer or an ink-jet printer controlled by a specialized CPU (not shown), which performs printing on a printing medium based on color image data or monochrome image data received from the printing controller 112. Further, an EEPROM (Electrically Erasable and Programmable Read Only Memory) (not shown) not requiring a backup power source is connected in the printing controller 112 for storing printing control parameters and the like.

A USB device controller 114, including a USB analog transceiver and a serial interface engine, performs communication control on USB interface. The USB device controller 114 performs protocol control in accordance with USB communication standards, converts data from a USB device control task executed by the CPU 101 into a packet and performs USB packet transmission to an external USB host unit (not shown) connected to the apparatus 100, on the other hand, converts a USB packet from an external USB host unit (not shown) connected to the apparatus 100 into data and transmits the data to the CPU 101. The USB communication

standards allow high-speed bidirectional data communication by 1 host (master) connected with plural devices (slaves). The USB device controller 114 has a device function in USB communication.

A USB host controller 115, including a USB analog transceiver and a serial interface engine, performs communication control on USB interface. The USB host controller 115 performs protocol control in accordance with the USB communication standards, converts data from a USB host control task executed by the CPU 101 into a packet and performs USB packet transmission to an external USB device unit (not shown) connected to the apparatus 100, on the other hand, converts a USB packet from an external USB device unit (not shown) connected to the apparatus 100 into data and transmits the data to the CPU 101. The USB communication standards allow high-speed bidirectional data communication by 1 host (master) connected with plural devices (slaves). The USB host controller 115 has a host function in USB communication.

A USB B connector 116 is a connector on the device side in conformity with the USB communication standards. It is a B type connector to be connected with an external USB host unit such as a personal computer.

The USB B connector 117 is a connector on the host side in conformity with the USB communication

standards. It is an A type connector to be connected with an external USB device unit such as a digital still camera.

The above constituent elements 101 to 106, 108 to 112, 114 and 115 are interconnected via a CPU bus 121 under the control of the CPU 101.

Next, the data arrangement in the ROM 102 and the image memory 104 of the information processing apparatus 100 will be described in detail.

Fig. 2 is an example of memory mapping showing storage areas of the ROM 102 and an image memory 104 of the information processing apparatus 100.

The entire nonvolatile area 201 corresponds to the ROM 102 of the information processing apparatus 100. The entire volatile area 202 corresponds to the image memory 104 of the information processing apparatus 100. First, the nonvolatile area 201 will be described in detail.

A program area 211 in the nonvolatile area 201 is a program area where control programs executed by the CPU 101, an operating system (OS) program and the like are stored. When the power of the information processing apparatus 100 is turned on, the CPU 101 of the information processing apparatus 100 starts its operation in accordance with program code stored in the program area 211.

A general-purpose driver area 212 in the

nonvolatile area 201 is a program area where a control program for the CPU 101 to control an external device connected to the USB A connector 117 of the information processing apparatus 100 in a general-purpose manner is stored. The control program to control the external device in a general-purpose manner means a control program to control an external device to realize communication in accordance with control standards standardized by a standardization organization, or de facto standard control standards.

For example, a function of downloading and print-outputting an image obtained by the information processing apparatus 100, a function of displaying a printing status on an external device connected to the apparatus, or the like, are attained by general-purpose control.

A data table area 213 in the nonvolatile area 201 is a data table area holding a data table referred to upon execution of processing by the CPU 101 of the information processing apparatus 100.

Next, the volatile area 202 will be described in detail. An image data area 214 in the volatile area 202 is an area where image data handled in the information processing apparatus 100, print data transmitted to the printing unit 113 of the information processing apparatus 100, status information obtained from the printing unit 113 and the like are stored. As

shown in Fig. 2, since the image data area 214 is dynamically ensured upon storage of image data or the like, plural image data areas 214 may exist separately.

A device-specific driver area 215 in the volatile area 202 is a device-specific driver area holding a device-specific driver for the CPU 101 of the information processing apparatus 100 to control an external device connected to the USB A connector 117 of the information processing apparatus 100, i.e., a digital still camera, in an expanded manner. The device-specific driver means a driver unique to the external device such as a digital still camera, made by its maker's own method. The device-specific driver cannot be utilized in general-purpose manner in digital still cameras of other makers. The device-specific driver corresponds to only the maker-specific or the device-specific function (for example, a function of enabling selection of printing mode unique to the maker and enable printing in the unique printing mode). Regarding general-purpose functions (e.g., downloading and print-outputting an image obtained by image sensing by the information processing apparatus 100), the device-specific driver may be used with a general-purpose driver. Further, it may be arranged such that all the functions provided from a digital still camera including the general-purpose functions are realized only by the device-specific driver.

As described later with reference to Fig. 3, since the device-specific driver area 215 is dynamically ensured upon reception and storage of device-specific driver from an external device connected to the USB A connector 117 of the information processing apparatus 100, plural device-specific driver areas 215 away from each other may exist.

A work area 216 in the volatile area 202 is a work area where work data or the like used by the CPU 101 of the information processing apparatus 100 for execution of software processing is stored.

Next, a control flow in a case where a digital still camera is connected to the information processing apparatus 100 and image data in the digital camera is print-outputted will be described in detail.

Fig. 3 is a flowchart showing an example of processing in a case where a digital still camera is connected to the information processing apparatus 100 and image data in the digital camera is print-outputted. Note that a description will be made on the assumption that the entire information processing apparatus 100 has been initialized by the CPU 101 prior to execution of the present flowchart and the information processing apparatus 100 is in a standby status waiting for operation start.

At step S301, the CPU 101 detects via the USB host controller 115 whether or not an external device,

i.e., a digital still camera, has been connected to the USB A connector 117. The detection as to whether or not a digital still camera has been connected to the USB A connector 117 is made by the CPU 101 using a mechanical hardware switch (not shown) via the USB host controller 115. Further, the detection of connection may be realized by the USB host controller 115 using a mechanical hardware switch (not shown) and the result of detection may be notified to the CPU 101. Further, the mechanical hardware switch may be replaced with an electrical state switch. Further, the mechanical hardware switch may be replaced with software data communication result.

If the CPU 101 determines at step S301 that a digital still camera has been connected to the USB A connector 117, the process flow proceeds to step S302. On the other hand, if the CPU 101 determines at step S301 that a digital still camera has not been connected to the USB A connector 117, the process flow repeats step S301.

At step S302, the CPU 101 starts control of the digital still camera connected to the USB A connector 117 in accordance with a general-purpose driver control program to control a general-purpose digital still camera stored in the ROM 102. The control of digital still camera is made as follows. First, communication specification information of the digital still camera

connected to the USB A connector 117 and digital still camera information are obtained. If the digital still camera is in conformity with the USB communication standards, information on the version of the USB communication standards, the USB class code, the communication-pipe packet size and the like can be obtained from the communication specification information. Further, information on the name of maker of the digital still camera, the product name and the like can be obtained from the digital still camera information. Next, the digital still camera connected to the USB A connector 117 is initialized. Thereafter, the process flow proceeds to step S303.

At step S303, the CPU 101 determines from the information obtained at step S302 whether or not the digital still camera connected to the USB A connector 117 has a device-specific driver to control the digital still camera. The determination as to whether or not the connected camera has a device-specific driver is made as follows. A table of names of the makers of digital still cameras having device-specific drivers and the product names is registered in a data table area 213 of the ROM 102 in the information processing apparatus 100 in advance. The CPU 101 compares the maker name and the product name obtained at step S302 with the data registered in the data table area 213 of the ROM 102, thereby determines whether or not the

digital still camera connected to the USB A connector 117 has a device-specific driver.

If the CPU 101 determines at step S303 that the digital still camera connected to the USB A connector 117 has a device-specific driver to control the digital still camera, the process flow proceeds to step S304.

Note that at step S303, it can be determined whether or not the connected camera supports "device-specific communication procedure" to be described later based on the maker name and the product name of the camera. If the camera supports the device-specific communication procedure, as a communication path is established, the device-specific driver can be reliably obtained.

If the CPU 101 determines at step S303 that the digital still camera connected to the USB A connector 117 does not have a device-specific driver to control the digital still camera, the process flow proceeds to step S308.

At step S304, the CPU 101 receives the device-specific driver from the digital still camera connected to the USB A connector 117, and determines whether or not the driver has been normally received. The reception of device-specific driver from the connected digital still camera is made as follows.

A digital still camera having a device-specific driver has a general-purpose device-specific driver

transmission mode for transmission of device-specific driver. The CPU 101 controls the USB host controller 115 to set the connected digital still camera in the device-specific driver transmission mode by the device-specific communication procedure. Then, the CPU 101 receives the device-specific driver from the connected digital still camera by the device-specific communication procedure.

If the CPU 101 has normally received the device-specific driver from the digital sill camera at step S304, the process proceeds to step S306. On the other hand, if the CPU 101 has not normally received the device-specific driver from the digital still camera at step S304, the process flow proceeds to step S305.

At step S305, the CPU 101 controls the operation display unit 108 to give a warning to the operator. More specifically, a message "Specialized driver to control connected device has not been received. Device operates by a lower-function standard driver." or the like is displayed on a display such as a LCD of the operation display unit 108, and the LED blinks. Thereafter, the process flow proceeds to step S308.

If the device-specific driver has been normally received at step S304, then at step S306, the device-specific driver obtained from the digital still camera at step S304 is stored in the device-specific driver area 215 of the image memory 104, and it is determined

whether or not the storage has been normally completed. The storage of device-specific driver into the device-specific driver area 215 of the image memory 104 is made as follows.

First, the CPU 101 checks an available area of the volatile area 202. The volatile area 202 includes the image data area 214 and the work area 216 as well as the device-specific driver area 215. The CPU 101 newly allocates an a unused area in the volatile area 202 to the device-specific driver area 215. Then the CPU 101 stores the device-specific driver obtained from the digital still camera at step S304 into the newly allocated device-specific driver area 215.

Further, the determination as to whether or not the storage of device-specific driver has been normally completed is made as follows.

When the CPU 101 checks an available area of the volatile area 202, if a large amount of image data has been received and the image data area 214 occupies the volatile area 202, the CPU 101 cannot ensure an available area of the volatile area 202 and cannot normally store the device-specific driver. In such case, it is determined that the storage of device-specific driver has not been normally completed.

Further, when the CPU 101 stores the device-specific driver obtained from the digital still camera at step S304 into the newly allocated device-specific

driver area 215, the CPU 101 writes the device-specific driver into the device-specific driver area 215, then reads the device-specific driver from the digital still camera again, and compares the written device-specific driver with the read device-specific driver. If some difference between both data has been detected, the CPU 101 also determines that the storage of device-specific driver has not been normally completed.

If the CPU 101 determines at step S306 that the storage of device-specific driver has been normally completed, the process flow proceeds to step S307. On the other hand, if the CPU 101 determines at step S306 that the storage of device-specific driver has not been normally completed, the process flow proceeds to step S305.

At step S307, the CPU 101 changes control of the digital still camera connected to the USB A connector 117, which was performed in accordance with a general-purpose driver control program, to control a general-purpose digital still camera, stored in the general-purpose driver area 212 of the ROM 102, to control performed in accordance with a control program of the device-specific driver stored at step S306 in the device-specific driver area 215 of the image memory 104. Thereafter, the process flow proceeds to step S308.

At step S308, if the operator performs a printing operation at the operation unit (not shown) of the

digital still camera connected to the USB A connector 117, the CPU 101 of the information processing apparatus 100 detects the printing operation, then receives image data from the digital still camera connected to the USB A connector 117. Generally, image data held in the digital still camera is stored in the form of JPEG compressed files in plural folders. Thereafter, the process flow proceeds to step S309.

At step S309, the CPU 101 processes the image data obtained from the digital still camera at step S308, outputs print data to the print controller 112, thereby performs printing processing. The image data processing performed by the CPU 101 includes the following processing.

First, as the image data received from the digital still camera is compressed by the JPEG method, the CPU 101 controls the coding/decoding processor 111 thereby decodes the JPEG-compressed file format data to raw data. Then the CPU 101 performs image processing such as color space processing on the raw image data in correspondence with the printing unit 113. Then the CPU 101 transfers the image-processed raw image data to the print controller 112. As a result, the image data is print-outputted to a printing medium in the printing unit 113.

In the above description, the data table area 213 is ensured in the ROM 102 as the nonvolatile area 201.

However, the present invention is not limited to this arrangement. For example, the information processing apparatus 100 may further have storage means as a data table memory in addition to the ROM 102 or the image memory 104. In this case, it is preferable that the data table memory is an electrically-rewritable storage means (e.g., EEPROM).

If the information processing apparatus has such data table memory, it can obtain latest information on the maker name, the product name and the like of a digital still camera having a device-specific driver via the communication line 131, and update the data table using the obtained information. In this arrangement, even if a new digital still camera having a latest function has been put on sale, a device-specific driver to realize the latest function can be obtained on the information processing apparatus 100 side.

Next, a description will be made in detail about a control flow in a case where a list of drivers stored in the general-purpose driver area 212 and the device-specific driver area 215 of the information processing apparatus 100 is displayed on the operation display unit 108, and a selectively-designated device-specific driver stored in the device-specific driver 215 is deleted.

Fig. 4 is a flowchart showing in detail a control

flow in a case where a list of drivers stored in the general-purpose driver area 212 and the device-specific driver area 215 of the information processing apparatus 100 is displayed on the operation display unit 108, and a selectively-designated device-specific driver stored in the device-specific driver 215 is deleted. Note that a description will be made on the assumption that the entire information processing apparatus 100 has been initialized by the CPU 101 prior to execution of the present flowchart and the information processing apparatus 100 is in a standby status waiting for operation start.

At step S401, the CPU 101 monitors the operation display unit 108, and determines whether or not an operation to display the list of drivers stored in the general-purpose driver area 212 and the device-specific driver area 215 has been performed by the operator. The operation to display the list of drivers by the operator may be made by using a specialized driver list display button (not shown) as a single-touch operation, otherwise, may be made by tracking a hierarchy from a menu button (not shown).

If it is determined at step S401 that an operation to display the list of stored drivers has been performed, the process flow proceeds to step S402. On the other hand, if it is determined at step S401 that an operation to display the list of stored drivers

has not been performed, the process flow repeats step S401.

At step S402, the CPU 101 obtains information on the general-purpose drivers stored in the general-purpose driver area 212 of the ROM 102. The general-purpose driver information means, e.g., the type of general-purpose driver, the version number, the date of generation, corresponding models, the data size and the like. Thereafter, the process flow proceeds to step S403.

At step S403, the CPU 101 controls the operation display unit 108, thereby displays the general-purpose driver information obtained at step S402 on the display unit such as a LCD of the operation display unit 108. Thereafter, the process flow proceeds to step S404.

At step S404, the CPU 101 obtains information on the device-specific drivers stored in the device-specific driver area 215 of the image memory 104. The device-specific driver means, the type of device-specific driver, the version number, the date of generation, the date of storage, corresponding models, the data size and the like. Thereafter, the process flow proceeds to step S405.

At step S405, the CPU 101 controls the operation display unit 108 thereby displays the device-specific driver information obtained at step S404 on the display unit such as LCD of the operation display unit 108.

Thereafter, the process flow proceeds to step S406.

At step S406, the CPU 101 monitors the operation display unit 108, and determines whether or not an operation to delete a device-specific driver displayed on the display unit such as LCD of the operation display unit 108 has been performed by the operator. The operation to delete a device-specific driver by the operator may be made by using a specialized device-specific driver deletion button (not shown) as a single-touch operation, otherwise, may be made by tracking a hierarchy from a menu button (not shown).

If it is determined at step S406 that an operation to delete a displayed device-specific driver has been performed, the process flow proceeds to step S407. On the other hand, if it is determined at step S406 that an operation to delete a displayed device-specific driver has not been performed, the process flow ends.

At step S407, the CPU 101 performs deletion of the device-specific driver designated by the operator at step S406. When the device-specific driver has been deleted from the device-specific driver area 215, the available area of the volatile area 202 is enlarged. Thereafter, the process flow ends.

In the present control flow, the list of drivers stored in the general-purpose driver area 212 and the device-specific driver area 215 of the information

processing apparatus 100 is displayed on the operation display unit 108, and a selectively designated device-specific driver stored in the device-specific driver area 215 is deleted. The control flow may be applied to a case where the list of drivers stored in the general-purpose driver area 212 and the device-specific driver area 215 of the information processing apparatus 100 is printed by the printing unit 113, and a selectively designated device-specific driver stored in the device-specific driver area 215 is deleted.

More specifically, the present control flow can be modified as a control flow in the case where the list of drivers stored in the general-purpose driver area 212 and the device-specific driver area 215 of the information processing apparatus 100 is printed by the printing unit 113, and a selectively designated device-specific driver stored in the device-specific driver area 215 is deleted, simply by replacing "list display operation" at step S401 with "list printing operation", replacing "display general-purpose driver information" at step S403 with "print general-purpose driver information", and replacing "display device-specific driver information" at step S405 with "print device-specific driver information".

As described, above, according to the present invention, a general-purpose external device can be reliably controlled in accordance with generalized

communication standards, on the other hand, when a special external device is connected, the external device can be controlled with an expanded function.

Further, a general-purpose external device can be reliably controlled in accordance with generalized communication standards.

Further, the device-specific drivers stored in the information processing apparatus 100 can be freely changed by the operator.

Further, it is not necessary to previously store all the device-specific drivers to control different external devices in the information processing apparatus 100. Further, it is possible for the operator to freely select any combination between the external devices and the information processing apparatus 100. Furthermore, even in the combination between an external device with a new function and an old information processing apparatus 100, it is possible to fully control the external device with new function by obtaining a device-specific driver corresponding to the new function. Accordingly, more reliable operations can be provided to the operator.

Further, since it is not necessary to re-transmit a unit-specific driver from external device unless data stored in the device-specific driver area 215 in the information processing apparatus 100 is deleted, it is possible for the operator to quickly perform a

processing operation.

Further, as the device-specific driver area 215 can be ensured, with other arbitrary area(s), in the image memory 104, effective utilization of the storage can be realized, and the cost of the information processing apparatus can be reduced.

Further, as the information processing apparatus 100 has means for giving a warning to the operator in a case where there is no area to dynamically store a device-specific driver in the image memory 104, the operation status can be notified to the operator, thereby the operator side can take a countermeasure. This also provides more reliable operations.

As described above, according to the present invention, an external device can be controlled in a general-purpose manner, and can be controlled with an expanded function in correspondence with the external device.

<Other Embodiments>

Note that the present invention can be applied to an apparatus comprising a single device or to system constituted by a plurality of devices.

Furthermore, the invention can be implemented by supplying a software program, which implements the functions of the foregoing embodiments, directly or indirectly to a system or apparatus, reading the supplied program code with a computer of the system or

apparatus, and then executing the program code. In this case, so long as the system or apparatus has the functions of the program, the mode of implementation need not rely upon a program.

Accordingly, since the functions of the present invention are implemented by computer, the program code installed in the computer also implements the present invention. In other words, the claims of the present invention also cover a computer program for the purpose of implementing the functions of the present invention.

In this case, so long as the system or apparatus has the functions of the program, the program may be executed in any form, such as an object code, a program executed by an interpreter, or script data supplied to an operating system.

Examples of storage media that can be used for supplying the program are a floppy disk, a hard disk, an optical disk, a magneto-optical disk, a CD-ROM, a CD-R, a CD-RW, a magnetic tape, a non-volatile type memory card, a ROM, and a DVD (DVD-ROM, DVD-R or DVD-RW).

As for the method of supplying the program, a client computer can be connected to a website on the Internet using a browser of the client computer, and the computer program of the present invention or an automatically-installable compressed file of the program can be downloaded to a recording medium such as

a hard disk. Further, the program of the present invention can be supplied by dividing the program code constituting the program into a plurality of files and downloading the files from different websites. In other words, a WWW (World Wide Web) server that downloads, to multiple users, the program files that implement the functions of the present invention by computer is also covered by the claims of the present invention.

It is also possible to encrypt and store the program of the present invention on a storage medium such as a CD-ROM, distribute the storage medium to users, allow users who meet certain requirements to download decryption key information from a website via the Internet, and allow these users to decrypt the encrypted program by using the key information, whereby the program is installed in the user computer.

Besides the cases where the aforementioned functions according to the embodiments are implemented by executing the read program by computer, an operating system or the like running on the computer may perform all or a part of the actual processing so that the functions of the foregoing embodiments can be implemented by this processing.

Furthermore, after the program read from the storage medium is written to a function expansion board inserted into the computer or to a memory provided in a

function expansion unit connected to the computer, a CPU or the like mounted on the function expansion board or function expansion unit performs all or a part of the actual processing so that the functions of the foregoing embodiments can be implemented by this processing.

As many apparently widely different embodiments of the present invention can be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the appended claims.